1. static void Main(string[] args)

{

int intSalary = 10000;

double dblBonus = 0;

if (intSalary < 10000)

{

dblBonus = 0;

}

else

{

dblBonus = 1000;

}

2. You cannot have an int statement and bool statement with &&.

3. static void Main(string[] args)

{

double dblBonus = 0;

int intSalary = 10000;

if (intSalary < 10000)

{

dblBonus = 0;

}

else

{

dblBonus = intSalary \* .2;

}

Console.WriteLine("Your bonus is "+dblBonus);

Console.ReadKey();

}

4. Cannot have two variables in the same scope income and income.

5.

static void Main(string[] args)

{

double dblBonus = 0;

double dblSales = 20000;

if (dblSales > 20000)

{

dblBonus = dblSales \*.1;

}

else

{

dblBonus = dblSales \* .02;

}

Console.WriteLine("Your bonus is "+dblBonus);

Console.ReadKey();

}

6. The variable state cannot be = to two values at the same time.

7.

static void Main(string[] args)

{

int intYears = 3;

double dblSales = 50000;

Boolean blnBonus;

if (intYears > 3 && dblSales > 50000)

{

blnBonus = true;

Console.WriteLine("Yes");

}

else

{

blnBonus = false;

Console.WriteLine("No");

}

Console.ReadKey();

}

8.

static void Main(string[] args)

{

double dblTax = 0;

string strState = "WA";

string strCity = "Seattle";

if (strState == "WA")

{

if (strCity == "Seattle")

{

dblTax = .09;

}

else if (strCity == "Spokane")

{

dblTax = .087;

}

else

{

dblTax = .084;

}

}

if (strState == "CA")

{

dblTax = .098;

}

Console.WriteLine(dblTax);

Console.ReadKey();

}

9. static void Main(string[] args)

{

string strState = "WA";

double dblTax = 0;

if (strState == "WA")

{

dblTax = .089;

}

else if (strState == "OR" || strState == "MT")

{

dblTax = 0;

}

else if (strState == "CA")

{

dblTax = .098;

}

else if (strState == "NV")

{

dblTax = .057;

}

else if (strState == "ID")

{

dblTax = .078;

}

else

{

dblTax = .06;

}

Console.WriteLine(dblTax);

Console.ReadKey();

}

10. static void Main(string[] args)

{

string strState = "WA";

double dblTax = 0;

switch (strState)

{

case "WA":

dblTax = .089;

break;

case "OR" || "MT":

dblTax = 0;

case "CA":

dblTax = .098;

case "NV":

dblTax = .057;

case "ID":

dblTax = .078;

case "default":

dblTax = .06;

break;

}

Console.WriteLine(dblTax);

Console.ReadKey();

}

11. || is a simple operator that is the same as saying “or” and will be true if something is true (something 1 || something 2) for example. Now if I was wanting to still have a similar result but I want to check if both are true I would write it like so (something | something1 | something3|). This would tell me if all are true while giving the same result of running my program. & is bit wise which evaluates both sides of the operation whereas && evaluates the left side of the operation to see if it is true. An example of using this would be…

int i = 0;

if (false & ++i == 1)

12. static void Main(string[] args)

{

int intAge = 18;

double dblRate = 0;

//intCr = Credit Rating

int intCr = 500;

bool blnCreditCard = true;

if (intAge >= 18 && intCr >= 500 && intAge < 24)

{

dblRate = .10;

}

if (intAge >= 24)

{

if (intCr >= 600 && intCr <= 700)

{

dblRate = .095;

}

else if (intCr >= 700 && intCr <= 800)

{

dblRate = .09;

}

else if (intCr >= 800)

{

dblRate = .0875;

}

else

{

dblRate = .10;

}

}

if (blnCreditCard && dblRate > 0)

{

dblRate += .05;

}

Console.WriteLine(dblRate);

Console.ReadKey();

}